



Universidade Federal de Pernambuco  
Centro de Informática  
Curso de Bacharelado em Engenharia da Computação

# **AndroidDriller:**

## **Uma ferramenta de mineração de repositórios Android**

Aluno: Alberto Vital Santos de Sousa  
Orientador: Leopoldo Motta Teixeira

Recife, junho de 2018

Universidade Federal de Pernambuco  
Centro de Informática  
Curso de Bacharelado em Engenharia da Computação

# **AndroidDriller:**

## **Uma ferramenta de mineração de repositórios Android**

Monografia apresentada ao Centro de Informática (CIn)  
da Universidade Federal de Pernambuco (UFPE), como requisito  
parcial para conclusão do Curso de Engenharia da Computação,  
orientada pelo professor Leopoldo Motta Teixeira.

Recife, junho de 2018

## **Agradecimentos**

# **1 Resumo**

Utilizando ferramentas capazes de minerar dados sobre repositórios, pesquisadores de engenharia de software têm obtido um melhor conhecimento sobre o processo de desenvolvimento de software, em geral. Este trabalho propõe uma ferramenta capaz de minerar repositórios fazendo uso das características comuns de projetos Android para que se possa realizar uma investigação preliminar de como as mudanças acontecem no desenvolvimento de aplicações para a plataforma.

## **2 Abstract**

# Sumário

<b>1</b>	<b>Resumo</b>	<b>1</b>
<b>2</b>	<b>Abstract</b>	<b>2</b>
<b>3</b>	<b>Introdução</b>	<b>4</b>
<b>4</b>	<b>Objetivos</b>	<b>5</b>
<b>5</b>	<b>Conceitos Básicos</b>	<b>6</b>
5.1	Android . . . . .	6
5.1.1	AndroidManifest . . . . .	7
5.1.2	Activity . . . . .	7
5.1.3	Broadcast Receiver . . . . .	7
5.1.4	Content Provider . . . . .	7
5.1.5	Service . . . . .	7
5.1.6	Permission . . . . .	7
5.2	Sistemas de controle de versão . . . . .	7
5.2.1	Git . . . . .	7
5.3	Mineração de repositórios . . . . .	7
5.4	RepoDriller . . . . .	7
<b>6</b>	<b>Trabalhos Relacionados / Contexto</b>	<b>8</b>
<b>7</b>	<b>AndroidDriller</b>	<b>9</b>
7.1	Metodologia . . . . .	9
7.2	Implementação . . . . .	9
7.3	Experimento . . . . .	9
7.4	Resultados . . . . .	9
<b>8</b>	<b>Conclusão</b>	<b>10</b>

### 3 Introdução

Em engenharia de software, a análise de repositórios tem ajudado pesquisadores a obter um melhor conhecimento sobre o processo de desenvolvimento de um software [1]. Com isso, é possível prever bugs e também analisar o padrão de desenvolvimento utilizado pelos colaboradores do projeto. Para tal fim, existem ferramentas capazes de minerar dados sobre repositórios, como por exemplo, o RepoDriller [2].

Ferramentas mais gerais, como o próprio RepoDriller, servem para analisar software em vários domínios. No desenvolvimento de aplicações Android, há diversas particularidades envolvidas. Por exemplo, o arquivo de manifesto contém informações sobre os vários componentes do projeto. Assim, as principais alterações feitas no repositório são refletidas em modificações desse arquivo.

No estudo de repositórios Android, pesquisadores têm usado ferramentas capazes de extrair dados dos arquivos do tipo apk [3], [4] e [5]. Para realizar estudos semelhantes, uma outra abordagem seria utilizar uma ferramenta capaz de minerar os repositórios das aplicações e extrair dados específicos levando em consideração não só o arquivo apk mas também as modificações dos arquivos de manifesto, por exemplo.

## **4 Objetivos**

Este trabalho tem por objetivo realizar uma investigação preliminar de como as mudanças acontecem no desenvolvimento de aplicações Android para criar ferramentas de suporte à evolução de aplicativos, e assim, implementar uma ferramenta capaz de minerar estes repositórios, aproveitando-se das características particulares de projetos Android.



## 5 Conceitos Básicos

### 5.1 Android

O sistema operacional Android pode ser definido como uma pilha de softwares de código aberto desenvolvida para sistemas móveis [6]. A partir desta pilha, programadores são capazes de desenvolver aplicações para dispositivos móveis sem se preocupar com detalhes específicos de hardware, pois a própria plataforma trata essas diferenças em sua API. O fato de Android ter seu código aberto, também permite ao desenvolvedor implementar sua própria versão dos módulos da plataforma caso deseje.

A pilha de software citada acima pode ser dividida em camadas, conforme a figura 5.1

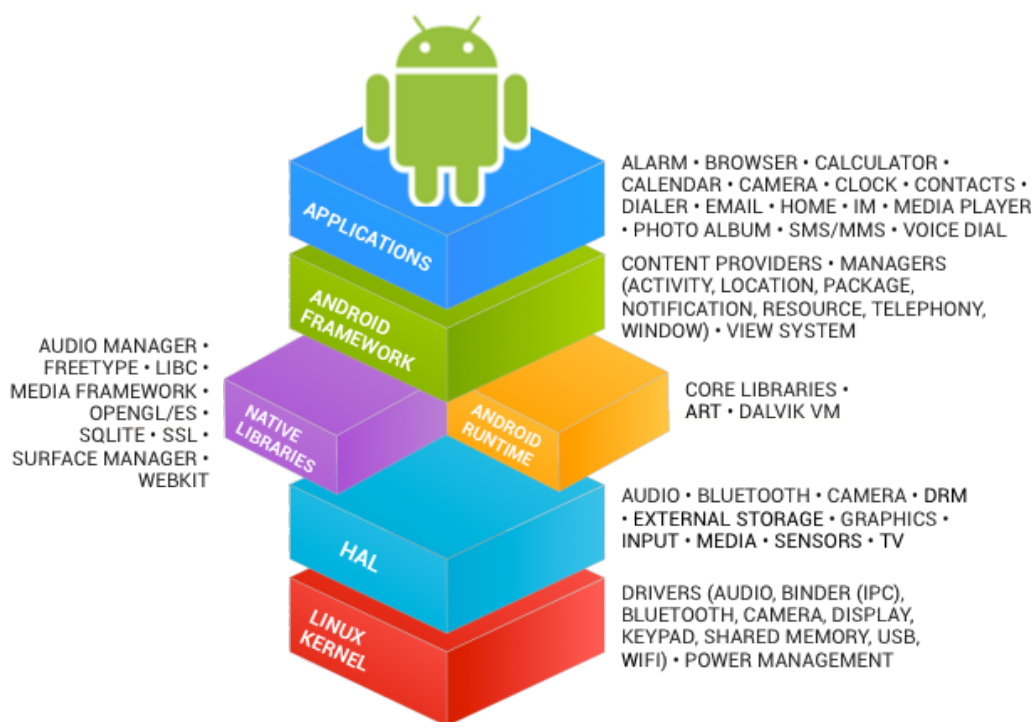


Figura 5.1: Pilha de software dividida em camadas [6]

Este trabalho propõe uma ferramenta para extrair dados relativos à camada *Android Framework*, onde estão os elementos de mais alto nível provido pela plataforma. Nesta camada estão as principais classes utilizadas pelos desenvolvedores

para produzir suas aplicações (que estão na camada logo acima). Mais abaixo trataremos alguns dos principais elementos da camada de *framework* e que são abordados pela ferramenta proposta.

### **5.1.1 AndroidManifest**

É um arquivo xml obrigatório que contém informações sobre os componentes da aplicação. Precisa ser criado com o nome *AndroidManifest.xml*, dessa forma o framework é capaz de identificar, dentre outras coisas, qual parte do código implementa a cada uma das entidades que compõem a aplicação.

### **5.1.2 Activity**

Representa a interface visual de interação com o usuário. No arquivo de manifesto é representada pela tag *activity* e precisa definir o atributo *name*, que é utilizado para identificar a classe que a implementa.

### **5.1.3 Broadcast Receiver**

Componente responsável

### **5.1.4 Content Provider**

### **5.1.5 Service**

### **5.1.6 Permission**

## **5.2 Sistemas de controle de versão**

### **5.2.1 Git**

## **5.3 Mineração de repositórios**

## **5.4 RepoDriller**

[2]

## **6 Trabalhos Relacionados / Contexto**

Em [3], [4], [5], etc. Foram feitos estudos tais que poderiam se aproveitar da ferramenta proposta.

## **7 AndroidDriller**

### **7.1 Metodologia**

Foi criada uma ferramenta Java que faz uso da API provida pelo RepoDriller para percorrer os commits do repositório. Neste projeto foram implementadas as classes capazes de colher dados sobre os componentes específicos de Android e geração de relatórios em arquivos CSV. Para visualização dos dados, foi implementado um programa na linguagem Python capaz de produzir gráficos a partir dos arquivos CSV gerados pela ferramenta.

### **7.2 Implementação**

Explicação mais detalhada de cada classe implementada

### **7.3 Experimento**

Detalhamento do experimento realizado na máquina do Cin

### **7.4 Resultados**

Análise dos resultados obtidos

## **8 Conclusão**

## Referências

- [1] C. Bird, P. C. Rigby, E. T. Barr, D. J. Hamilton, D. M. German, and P. Devanbu, “The promises and perils of mining git,” in *2009 6th IEEE International Working Conference on Mining Software Repositories*, pp. 1–10, May 2009.
- [2] “Repodriller.” <https://github.com/mauricioaniche/repodriller>.
- [3] P. Calciati and A. Gorla, “How do apps evolve in their permission requests?: A preliminary study,” in *Proceedings of the 14th International Conference on Mining Software Repositories, MSR ’17*, (Piscataway, NJ, USA), pp. 37–41, IEEE Press, 2017.
- [4] D. E. Krutz, N. Munaiah, A. Peruma, and M. W. Mkaouer, “Who added that permission to my app?: An analysis of developer permission changes in open source android apps,” in *Proceedings of the 4th International Conference on Mobile Software Engineering and Systems, MOBILESoft ’17*, (Piscataway, NJ, USA), pp. 165–169, IEEE Press, 2017.
- [5] Y. Lyu, J. Gui, M. Wan, and W. G. J. Halfond, “An empirical study of local database usage in android applications,” in *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 444–455, Sept 2017.
- [6] “The android source code.” <https://source.android.com/setup/>.